Image Annotation with TagProp on the MIRFLICKR set

Jakob Verbeek, Matthieu Guillaumin, Thomas Mensink, Cordelia Schmid

Image Annotation with TagProp on the MIRFLICKR Set

Jakob Verbeek*
INRIA Rhône-Alpes
655 Avenue de l'Europe
38330 Montbonnot, France

Matthieu Guillaumin
INRIA Rhône-Alpes
655 Avenue de l'Europe
38330 Montbonnot, France

Thomas Mensink
Xerox Research Centre Europe
6 chemin de Maupertuis
38240 Meylan, France

Cordelia Schmid
INRIA Rhône-Alpes
655 Avenue de l'Europe
38330 Montbonnot, France

ABSTRACT

Image annotation is an important computer vision problem where the goal is to determine the relevance of annotation terms for images. Image annotation has two main applications: (i) proposing a list of relevant terms to users that want to assign indexing terms to images, and (ii) supporting keyword based search for images without indexing terms, using the relevance estimates to rank images.

In this paper we present TagProp, a weighted nearest neighbour model that predicts the term relevance of images by taking a weighted sum of the annotations of the visually most similar images in an annotated training set. TagProp can use a collection of distance measures capturing different aspects of image content, such as local shape descriptors, and global colour histograms. It automatically finds the optimal combination of distances to define the visual neighbours of images that are most useful for annotation prediction. TagProp compensates for the varying frequencies of annotation terms using a term-specific sigmoid to scale the weighted nearest neighbour tag predictions.

We evaluate different variants of TagProp with experiments on the MIR Flickr set, and compare with an approach that learns a separate SVM classifier for each annotation term. We also consider using Flickr tags to train our models, both as additional features and as training labels. We find the SVMs to work better when learning from the manual annotations, but TagProp to work better when learning from the Flickr tags. We also find that using the Flickr tags as a feature can significantly improve the performance of SVMs learned from manual annotations.

*Author email addresses are firstname.lastname@inria.fr.
We would like to thank the ANR project R2I as well as the QUAERO project for their financial support.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Image annotation

1. INTRODUCTION

In image retrieval the goal is to find images from a database that are relevant to a user specified query. The two main image retrieval scenarios are query by example, and keyword-based queries. In the former, the user gives a query image and the goal is to retrieve 'similar' images from the database. Image similarity can be defined in several ways, but generally the goal is to find images of the same scene or the same objects, while being robust against changes in the imaging conditions (e.g. changes in viewpoint, scale, lighting, occlusions, etc.), see e.g. [12]. For the latter, the goal is to retrieve images that are relevant to the query keywords. In this case the images in the database should be indexed with the terms in an annotation vocabulary. Obviously, manually assigning keywords to images is a tedious process, hence the interest in automatically predicting the annotation term relevance for images, see e.g. [8, 15, 16, 19].

There are two ways in which we can use automatic image annotation to facilitate keyword-based image search. First, we can assist a user to annotate his images by proposing a short list of annotation terms sorted by their estimated relevance. This is advantageous if the vocabulary contains many words (say more than 100), allowing the user to quickly select the relevant terms without exhaustively checking the complete list of terms. Second, the relevance predictions can be used directly to enable keyword-based search on image collections that lack manual annotations. In this case, we rank the images by the likelihood that their annotation contains the query terms.

In this paper we present TagProp, a weighted nearest neighbour image annotation model for which the parameters are set by maximising the likelihood of the annotations of a collection of annotated training images. Annotation terms of

clouds	sky (0.99)
sea	clouds (0.94)
sky	water (0.90)
structures	sea (0.70)
sunset	sunset (0.51)
water	structures (0.43)

female	people (0.62)
indoor	indoor (0.49)
male	female (0.31)
night	portrait (0.30)
people	male (0.24)
portrait	night (0.13)

clouds	sky (0.60)
female	structures (0.36)
male	tree (0.24)
people	people (0.18)
sky	clouds (0.17)
structures	indoor (0.13)

clouds	sky (0.99)
male	clouds (0.99)
people	water (0.69)
sea	structures (0.64)
sky	sea (0.32)
water	tree (0.32)

animals	sky (0.90)
bird	water (0.53)
lake	clouds (0.45)
river	structures (0.39)
sea	transport (0.29)
water	sunset (0.22)

animals	sky (0.52)
bird	water (0.50)
lake	structures (0.48)
river	people (0.23)
sea	tree (0.22)
water	clouds (0.20)

Figure 1: Example images from the MIR Flickr set, for each image we show the manually assigned annotation terms, and those predicted using TagProp with the relevance estimate in brackets, and underlined if correct. In the first row the top predicted terms coincide with the actual predictions, the middle row four of the top six terms are correct (a typical situation), and in the last row only one of the top six predictions is correct.

test images are predicted by means of a weighted sum of the annotations of their neighbours: the visually most similar images in the training set. TagProp can combine a collection of several distance measures to define visual similarity, capturing different aspects of image content, such as local shape descriptors, and global colour histograms. The parameters of the model combine the various visual similarities to define the optimal weights to training images in terms of the likelihood criterion. TagProp also includes a term-specific sigmoid function to compensate for the different frequencies of annotation terms.

Our model is inspired by recent successful methods [6, 13, 18], that propagate the annotations of training images to new images. Our models are learnt in a discriminative manner, rather than using held-out data [6], or using neighbours in an adhoc manner to annotate images as in [18]. In [18] the authors also tried to combine different image similarities by learning a binary classifier separating image pairs that have several tags in common from images that do not share any tags. However, this approach did not give better results than an equally weighted combination of the distance measures. Our model does successfully combine different similarity measures, because we integrate learning the distance combination in the model, rather than learning it through solving an auxiliary problem. Other nearest neighbour techniques for image annotation include methods based on label diffusion over a similarity graph of labeled and unlabelled images [16, 22], or learning discriminative models in neighbourhoods of test images [25].

Other related work includes a variety of generative models. To annotate a new image these models compute the conditional probability over annotation terms given the visual features of the image. One important family of methods is based on topic models such as latent Dirichlet allocation, probabilistic latent semantic analysis, and hierarchical Dirichlet processes, see e.g. [1, 20, 24]. A second family of methods uses mixture models to define a joint distribution over image features and annotation tags. Sometimes a fixed number of mixture components over visual features per keyword is used [3], while other models use the training images as components to define a mixture model over visual features and tags [6, 13]. The latter can be seen as non-parametric density estimators over the co-occurrence of images and annotations. A potential weakness of generative models is that they maximise the generative data likelihood, which is not necessarily optimal for predictive performance. Discriminative models for tag prediction have also been proposed [4, 8, 10]. These methods learn a separate classifier for each annotation term to predict whether a test image is relevant.

We assess the image annotation performance of different variants of TagProp, and compare against an approach that learns a separate classifier for each annotation term to predict its relevance for an image. For the separate classifiers we choose non-linear support vector machines (SVMs) based on local image features, which have shown state-of-the-art performance for image classification [26]. Our evaluations are performed using the MIR Flickr set [11], a recent data set that contains 25.000 images downloaded from the Flickr

photo sharing website[1]. For each image, the tags associated with the image on the Flickr website are available, as well as a precise manual annotation for 24 concepts. In Figure 1 we show several example images from the database, together with their manual annotations, and the annotations predicted using TagProp.

In our experiments we consider learning models from both the manual annotation and the Flickr tags, furthermore we consider the Flickr tags as additional features rather than training labels. When learning from the manual annotations we find that the SVM approach performs better than TagProp, albeit at the cost of learning separate models for each concept. The Flickr tags provide a strong additional feature for the SVM models in this case. When training the models on the basis of the Flickr tags instead of the manual annotation we find that TagProp gives best performance, probably because it has fewer parameters and is less likely to overfit to the noise in the user tags.

The rest of this paper is organized as follows. In the next section we present our TagProp model in detail. In Section 3 we describe the experimental setup, and present results in Section 4. We present our conclusions in Section 5.

2. IMAGE ANNOTATION WITH TAGPROP

In this section we first present TagProp, our weighted nearest neighbour annotation model. We assume that some visual similarity or distance measures between images are given, abstracting away from their precise definition. In Section 2.2 and Section 2.3 we proceed by discussing two ways to define the weights for neighbours in this model. In Section 2.4 we extend the model by adding a per-word sigmoid function that can compensate for the different frequencies of annotation terms in the database.

2.1 A Weighted Nearest Neighbour Model

In the following we use $y_{iw} \in \{-1, +1\}$ to denote whether concept w is relevant for image i or not. The probability that concept w is relevant for image i, i.e. $p(y_{iw} = +1)$, is obtained by taking a weighted sum of the relevance values for w of neighbouring training images j. Formally, we define

$$p(y_{iw} = +1) = \sum_j \pi_{ij} p(y_{iw} = +1|j), \tag{1}$$

$$p(y_{iw} = +1|j) = \begin{cases} 1 - \epsilon & \text{for } y_{jw} = +1, \\ \epsilon & \text{otherwise.} \end{cases} \tag{2}$$

The π_{ij} denote the weight of training image j when predicting the annotation for image i. To ensure proper distributions, we require that $\pi_{ij} \geq 0$, and $\sum_j \pi_{ij} = 1$. Each term $p(y_{iw} = +1|j)$ in the weighted sum is the prediction according to neighbour j. Neighbours predict with probability $(1-\epsilon)$ that image i has the same relevance for concept w as itself. The introduction of ϵ is a technicality to avoid zero prediction probabilities when none of the neighbours j have the correct relevance value. In practice we fix $\epsilon = 10^{-5}$, although the exact value has little impact on performance.

The parameters of the model, which we will introduce below, control the weights π_{ij}. To estimate the parameters we maximize the log-likelihood of predicting the correct annotations for training images in a leave-one-out manner. Taking care to exclude each training image as a neighbour of itself, i.e. by setting $\pi_{ii} = 0$, our objective is to maximize

$$\mathcal{L} = \sum_{i,w} \ln p(y_{iw}). \tag{3}$$

Below, we discuss two different ways to define the weights of the model. Given a particular definition of the weights, the log-likelihood can be optimised using gradient descend.

2.2 Rank-based weighting

When using rank-based weights we set $\pi_{ij} = \gamma_k$ if j is the k-th nearest neighbour of i. This directly generalizes a simple K nearest neighbour approach, where the K nearest neighbours receive an equal weight of $1/K$. The data log-likelihood (3) is concave in the parameters γ_k, and can be maximised using an EM-algorithm or a projected-gradient algorithm. In our implementation we use the latter because of its speed. To limit the computational cost of the learning algorithm we only allow non-zero weights for the first K neighbours, typically K is in the order of 100 to 1000. The number of parameters of the model then equals K. By precomputing the K nearest neighbours of each training image the run-time of the learning algorithm is $O(NK)$ with N the number of training images. In Section 4 we show an example of a set of weights learned in this manner.

In order to make use of several different distance measures between images we can extend the model by introducing a weight for each combination of rank and distance measure. For each distance measure d we define a weight π_{ij}^d that is equal to γ_{dk} if j is the k-th neighbour of i according to the d-th distance measure. The total weight for an image j is then given by the sum of weights $\pi_{ij} = \sum_d \pi_{ij}^d$ obtained using different distance measures. Again we require all weights to be non-negative and to sum to unity: $\sum_{j,d} \pi_{ij}^d = 1$. In this manner we effectively learn rank-based weights per distance measure, and at the same time learn how much to rely on the rank-based weights provided by each distance measure.

2.3 Distance-based weighting

Alternatively, we can define the weights directly as a function of distance, rather than rank. In this case the weights will depend smoothly on the distance, and we can learn a distance measure that leads to optimal predictions. Here, we define the weights of training images j for an image i to decrease exponentially with distance by setting

$$\pi_{ij} = \frac{\exp(-d_{\boldsymbol{\theta}}(i,j))}{\sum_{j'} \exp(-d_{\boldsymbol{\theta}}(i,j'))}, \tag{4}$$

where $d_{\boldsymbol{\theta}}$ is a distance metric with parameters $\boldsymbol{\theta}$ that we want to optimize. Choices for $d_{\boldsymbol{\theta}}$ include Mahalanobis distances parametrized by a semi-definite matrix, and positive linear distance combinations $d_{\boldsymbol{\theta}}(i,j) = \boldsymbol{\theta}^\top \boldsymbol{d}_{ij}$ where $\boldsymbol{d}_{ij}$ is a vector of base distances between image i and j, and $\boldsymbol{\theta}$ contains the positive coefficients of the linear distance combination. In our experiments we have used the latter case of linear distance combinations, in which the number of parameters equals the number of base distances that are combined. When we use a single distance $\boldsymbol{\theta}$ is a scalar that controls the decay of the weights with distance, and it is the only parameter of the model. We maximize the log-likelihood using

[1]See `http://www.flickr.com`.

a projected gradient algorithm under positivity constraints on the elements of $\boldsymbol{\theta}$.

As with rank-based weights, we only compute weights for a limited number of K neighbours to reduce the computational cost of training the model. When using a single distance measure we simply select the K nearest neighbours, assuming that the weights will tend to zero for further neighbours. When learning a linear combination of several distances it is not clear beforehand which will be the nearest neighbours, as the distance measure changes during learning. Given that we will use K neighbours, we therefore include as many neighbours as possible from each base distance. In this way we are likely to include all images with large π_{ij} regardless of the distance combination $\boldsymbol{\theta}$ that is learnt.

2.4 Word-specific Logistic Discriminants

The weighted nearest neighbour model introduced above tends to have relatively low recall scores for rare annotation terms. This effect is easy to understand as in order to receive a high probability for the presence of a term, it needs to be present among most neighbours with a significant weight. This, however, is unlikely to be the case for rare annotation terms. Even if some of the neighbours with significant weight are annotated with the term, we will still tend to predict it with a low probability as compared to the predictions for frequent terms.

To overcome this, we introduce word-specific logistic discriminant model that can boost the probability for rare terms and possibly decrease it for frequent ones. The logistic model uses weighted neighbour predictions by defining

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w), \quad (5)$$

$$x_{iw} = \sum_j \pi_{ij} p(y_{iw} = +1|j), \quad (6)$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the sigmoid function, and x_{iw} is the weighted nearest neighbour prediction for term w and image i used before, c.f. Equation (1). The word-specific models adds two parameters per annotation term.

In practice we estimate the parameters $\{\alpha_w, \beta_w\}$ and the ones which control the weights in an alternating fashion. For fixed π_{ij} the model is a logistic discriminant model, and the log-likelihood is concave in $\{\alpha_w, \beta_w\}$, and can be trained per term. In the other step we optimize the parameters that control the weights π_{ij} using gradient descend. We observe rapid convergence, typically after alternating the optimization three times. We refer to [9] for the derivatives of the different variants of the model.

3. EXPERIMENTAL SETUP

In this section we describe the data set used in our experiments, the performance evaluation measures, and the visual feature extraction procedures.

3.1 The MIR Flickr set

The MIR Flickr set has been recently introduced [11] to evaluate keyword-based image retrieval methods. The set contains 25.000 images that were downloaded from the Flickr website. For each image the tags that Flickr users assigned to the image are available, as well as EXIF information fields. The tags are a valuable resource, but they tend to be unreliable. Not all tags are actually relevant to the image content, as users assign labels to several images at a time, and assign labels of objects or places even if they are not actually shown in the image. For example, images labeled with car might be taken from a car, rather than depicting one. Moreover, the user tags tend to be far from complete: usually people add a few tags, rather than an exhaustive list of relevant terms. In our experiments we limited the set of tags to the 457 most frequent ones that appear at least 50 times among all 25.000 images.

The images are also manually annotated for 24 concepts by asking people for each image whether it is at least partially relevant for each concept. A second round of annotation was performed for 14 concepts where a stricter notion of relevance was used. Here only images labeled as relevant for a concept in the first round were considered, and marked relevant only if a significant portion of the image is relevant for the concept. Throughout this paper we use a '*' to refer to the strict annotation for concepts, .e.g. 'baby*' refers to the strict annotation for 'baby'. In total each image is thus annotated by its relevance for 38 concepts. See Figure 1 for example annotations, and Table 1 for a list of the concepts.

To estimate and evaluate our models we have split the data set into equally sized train and test sets, by including every second image in the train set and using the remaining ones for the test set. In our experiments we have used both the annotation labels and the Flickr tags to learn our models. Because of the noise in the tag absence/presence, performance evaluation is always based on the manual annotations.

3.2 Performance Measures

To measure performance we use average precision (AP) and break even point precision (BEP). To compute these for a given semantic concept, we rank all images by predicted relevance and evaluate precision at each position, i.e. at position n we compute the fraction of images up to rank n that are indeed relevant according to the manual annotation. AP averages the precision over all positions of relevant images, whereas BEP computes the precision at position k, where k is the number of relevant images for the concept according to the manual annotation. Both measures are evaluated per concept, and possibly averaged over different concepts to obtain a single measure. These measures indicate how well we can retrieve images from the database in response to a keyword-based user query.

In addition to these per-concept measures, we also compute per-image measures as follows. For each image, we rank the concepts by their predicted relevance, and then compute AP and BEP as before. These per-image measures, which we denote iAP and iBEP respectively, indicate how well we can automatically identify relevant concepts for an image, e.g. to propose a list of relevant annotation terms to a user.

3.3 Visual Feature Extraction

For each image we extract features that are commonly used for image search and categorisation. We use two types of global image descriptors: Gist features [21], and colour histograms with 16 bins per colour channel, yielding $16^3 = 4096$

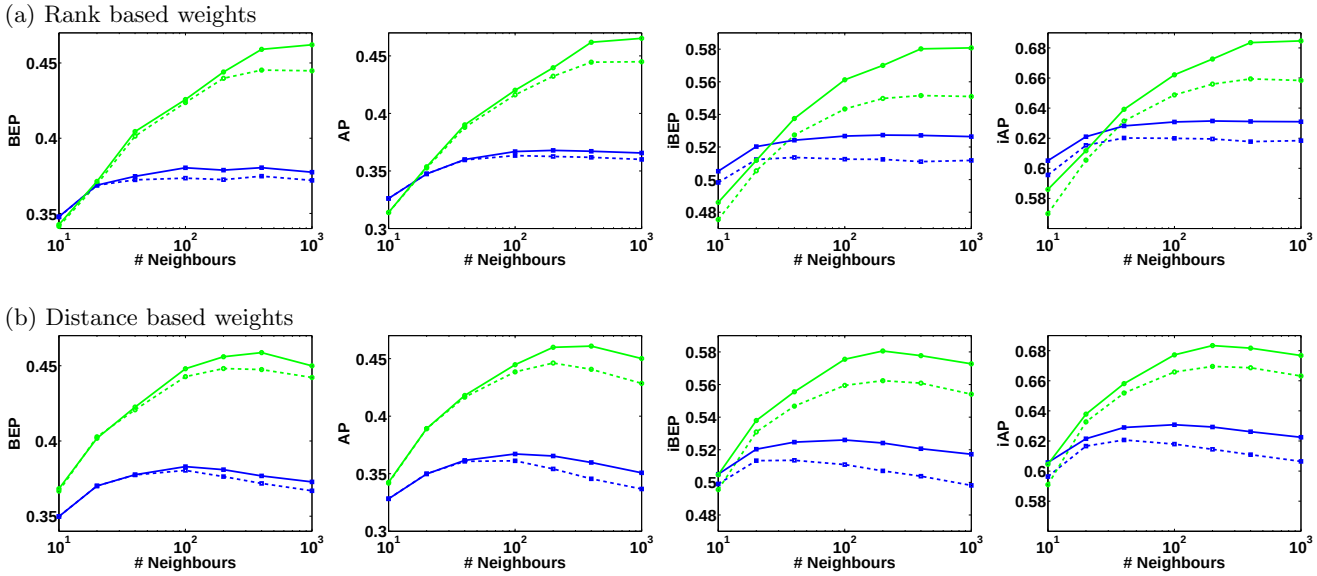

Figure 2: Performance of TagProp using (a) rank based weights, and (b) distance based weights. The TagProp variants either use a single distance (blue/dark curves) or all distances (green/light curves), and either include the sigmoid transformation (solid curves) or not (dashed curves). Note the log scale on the horizontal axis.

dimensional histograms. The colour histograms are computed in three spaces: RGB, LAB, and HSV. As local features we use the SIFT descriptor [17], as well as a robust hue descriptor [23]. Both are computed for regions on a dense multi-scale grid, and regions found using a Harris-Laplacian detector. Each local feature descriptor is quantized using k-means on samples from the training set, and images are represented as a 'bag-of-words' histogram. We used k-means with 1000 and 100 cluster centres for the SIFT and Hue descriptors respectively. Our histogram features, all except Gist, are L1 normalised.

Note that our histogram features are invariant to the layout of the image. To encode the spatial layout of the image to some degree, we follow the approach of [14], and compute the histograms over different image regions. We do so only for our histogram features, as the Gist already encodes some form of the layout of the image. We compute the histograms over three horizontal regions of the image, reflecting the typical layout of landscape photography. The three histograms are then concatenated to form a new global descriptor, albeit one that encodes some of the spatial layout of the image. To limit the size of the colour histogram, we reduced the quantization to 12 bins in each channel here. We use these new features in addition to the image-wide histograms. This results in 15 distinct features, namely one Gist descriptor, 6 colour histograms (3 colour spaces $\times$ 2 layouts) and 8 bag-of-features (2 descriptors $\times$ 2 detectors $\times$ 2 layouts). To compute distances from the descriptors we use L2 for Gist, L1 for colour histograms, and χ^2 for the others.

Some of the SVM and TagProp models in our experiments are based on a single distance. In this case we use an equally weighted sum of all 15 distance measures, which are all normalised to have a maximum value of 1. When using TagProp with rank-based weights and multiple distances, we also use the equally weighted sum of distances to define a 16-th set of neighbours. For distance-based weights it is not useful to include the equally weighted sum, as it is already a linear distance combination itself.

4. EXPERIMENTAL RESULTS

In this section we present our experimental results. In Section 4.1, we analyse the performance of TagProp when learning from the manual ground truth annotations, and in Section 4.2 we compare these results to learning an SVM classifier per concept. Finally, in Section 4.3 we consider the Flickr tags to learn the models.

4.1 Evaluating TagProp variants

In our first set of experiments we use different variants of TagProp to predict the relevance of the 38 manually annotated concepts. The variants of TagProp we included use rank-based or distance-based weights, optionally include the sigmoid transformation, and either use a single or multiple distance measures between images. In Figure 2 we give an overview of performance of the TagProp variants in terms of BEP, AP, iBEP, and iAP, as a function of the number of neighbours K that was used to train the model.

For both choices of weights we can observe that the sigmoid transformation of the predictions consistently has a beneficial effect. The effect is more pronounced in terms of iAP and iBEP than in AP and BEP. This is as expected since the sigmoid introduces a monotonic transform of the relevance estimates for a given concept. Therefore the ranking of images for a given concept is not affected much, and so similar AP and BEP values are obtained. However, for a particular image the sigmoid parameters for different classes can change the order of their relevance scores, and thus have a significant impact on the iBEP and iAP scores.

Using either rank-based or distance-based weights, we ob-

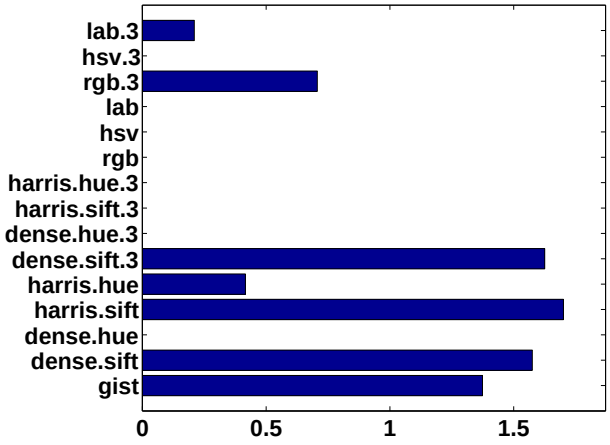

Figure 3: Coefficients of the linear distance combination learned with TagProp with distance-based weights, and sigmoid transformation included.

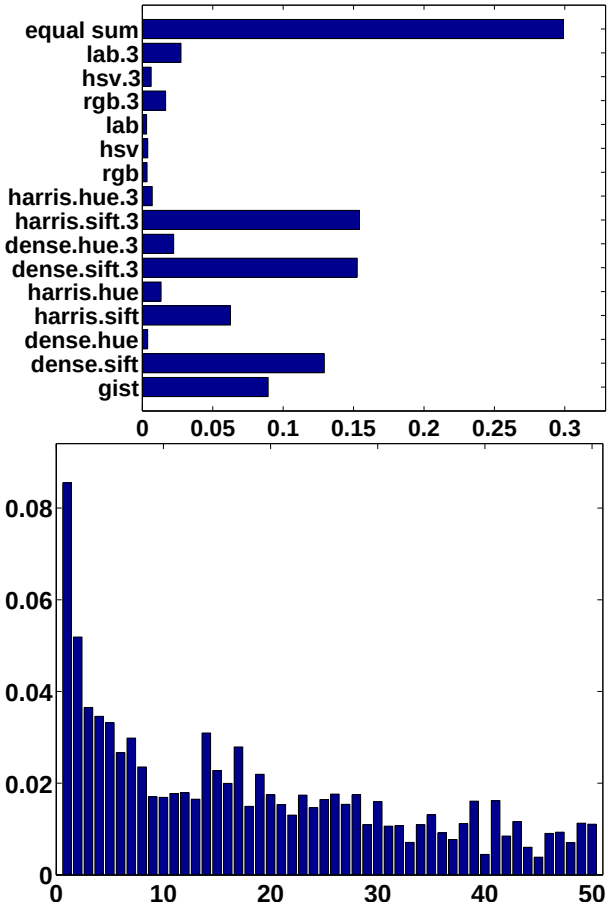

Figure 4: Rank-based weights learned using the 15 base distances and their equally weighted sum. The top panel shows the total weight associated with each distance, and the bottom panel shows the total weight associated with rank 1 up to 50.

serve that the performance can be significantly increased when learning how to combine the distances. The improvements are strongest when a relatively large number of neighbours is used. This is easily understood as in that case it is more likely to include useful neighbours from the different neighbourhoods. Note that when using rank-based weights and $K = 1000$ neighbours, we only have 62 neighbours for each of our distances. When using distance-based weights we can use K unique neighbours, and generally around $K = 200$ to $K = 500$ neighbours leads to optimal results. It is not quite clear why using more neighbours has a slight negative effect on performance; apparently the exponential decay of the weights with distance is appropriate only in moderately sized neighbourhoods. When using the fixed distance, about $K = 100$ neighbours is sufficient to obtain near optimal performance for both weight definitions.

To gain insight into which image distances are most important in the learned models, we show in Figure 3 the coefficients learned with the model using distance-based weights. Note that the weights are sparse: only seven of the 15 distance measures receive a non-zero weight. The most important distance measures are the ones based on the Gist descriptor, and the local SIFT descriptors. From the colour features, only the Harris-Hue and the LAB and RGB histograms that include spatial layout are used.

In Figure 4 we consider rank-based weights, when using the 15 base distances together with their equal sum to define a 16-th set of neighbours. Remember that in this case a weight for each combination of rank and distance measure is learned. To visualize the weights we look at the total weight assigned to neighbours of a certain distance, by summing over the weights assigned to that distance for different ranks. Similarly, we look at the total weight assigned to neighbours of a certain rank, by summing over distance measures. We observe that the weights drop quickly as a function of their rank, and that also in this case the Gist descriptor and the local SIFT descriptors are the most useful to define the weights of neighbours. Interestingly, the equal sum of distances receives the largest weight. This suggests that images that are similar according to multiple distance measures are the most useful to predict the annotations. However, by also assigning weight to neighbours from other distance measures a significant increase in performance is obtained, cf. Figure 2.

For the following experiments, we use TagProp models with sigmoid included, and with 200 and 1000 neighbours for distance-based and rank-based weights respectively. Figure 5 gives an overview of the most 'difficult' images for TagProp using distance-based weights: for each of the 14 concepts with a strict labeling we show the positive image with the lowest score, and the negative image with the highest score. Interestingly, for several concepts the highest scoring negative example can be argued to be actually relevant (e.g. for clouds, flower, night, portrait, river, sea, and tree).

4.2 Comparison with SVM classifiers

When dealing with a limited number of annotation concepts, we can learn a separate classifier for each one of them instead of using nearest neighbour style models as presented above. The advantage of such an approach is that a separate set

	animals	baby	baby*	bird	bird*	car	car*	clouds	clouds*	dog	dog*	female	female*
Distance	41.1	13.1	14.8	13.6	17.1	31.1	48.9	74.6	66.2	22.3	24.1	52.0	47.9
Rank	42.6	11.8	16.3	14.8	18.2	30.9	47.3	74.4	67.4	23.3	25.6	52.4	49.3
SVM	**48.6**	**13.3**	**18.9**	**20.3**	**22.7**	**34.6**	**50.2**	**84.8**	**77.4**	**29.7**	**33.6**	**56.3**	**53.8**
Random	12.9	1.1	0.5	3.0	2.0	5.0	1.7	14.5	5.4	2.7	2.3	24.8	15.9
	flower	flower*	food	indoor	lake	male	male*	night	night*	people	people*	plant	portrait
Distance	43.2	49.2	45.8	69.8	24.2	47.4	36.1	58.9	55.1	74.2	67.4	74.0	56.8
Rank	43.2	50.1	44.8	69.2	24.2	47.8	37.7	59.7	53.6	74.8	68.6	74.6	58.8
SVM	**53.4**	**63.8**	**48.9**	**75.0**	**26.9**	**50.2**	**41.8**	**65.5**	**55.2**	**79.4**	**75.6**	**79.6**	**68.4**
Random	7.4	4.4	4.0	33.5	3.0	23.9	14.2	10.3	2.5	41.3	31.1	34.8	15.6
	portrait*	river	river*	sea	sea*	sky	struct.	sunset	transp.	tree	tree*	water	Mean
Distance	56.3	21.5	5.8	50.4	20.2	84.5	76.3	57.7	42.0	59.7	43.2	56.7	45.9
Rank	58.6	23.6	6.5	50.3	24.4	84.5	76.1	57.8	42.4	60.1	41.6	57.7	46.5
SVM	**68.4**	**24.4**	**6.6**	**56.4**	**30.3**	**89.0**	**78.0**	**67.7**	**44.7**	**67.8**	**54.6**	**61.8**	**52.0**
Random	15.1	3.7	0.6	5.3	0.8	31.0	40.4	8.4	11.9	18.3	2.7	13.1	12.3

Table 1: Comparison in terms of AP of TagProp with distance-based and rank-based weights, and SVMs. Results for all 38 concepts are given, as well as their mean (last column).

of parameters can be learned for each concept to optimally separate the relevant from the non-relevant images.

We trained support vector machine (SVM) classifiers using RBF kernels based on the equally weighted sum of our base distances. The kernel function that compares two images is thus given by $k(x_i, x_j) = \exp(-d(x_i, x_j)/\lambda)$, where $d(x_i, x_j)$ is the equally weighted distance combination, and λ is set as the average of all pairwise distances among the training images. For a given concept, we can then rank the images by the classifier output score.

In order to rank the concepts for a given image we need to compare the SVM scores of different concepts. To this end we used 10% of the training data of each concept to learn a sigmoid to map the SVM scores to probabilities. In order to set the regularization parameter of the SVMs we perform 10 fold cross-validation.

In Table 1 we present AP scores per annotated concept for the SVM classifiers, as well as TagProp with distance-based and rank-based weights. For reference, we also included the precision for a random ranking, i.e. the fraction of relevant images per concept. On average, TagProp performs similar using either distance-based or rank-based weights, although for some concepts the scores differ up to 4% in terms of AP. For all concepts, the SVM approach yields higher AP scores than those obtained with TagProp, on average leading to a 5.5% higher AP score. In terms of BEP similar results are obtained, in this case SVMs score lower than TagProp for some classes, but on average SVMs still yield a 4.2% higher score. When assessing annotation performance per image, we find similar results. For SVMs we measured an average iBEP of 61.7% and iAP of 71.9%, while for distance-based TagProp we found an average iBEP of 58.1% and iAP of 68.3%, and TagProp using rank-based weights yielded 58.1% and 68.5% respectively.

Of course, the higher performance of the SVM approach comes at the cost of training a separate classifier per concept. Including the 10-fold cross-validation over 5 values of the regularization parameter this means we have to train $38 \times 10 \times 5 = 1900$ SVM classifiers over 10125 training images each (we use 90% of the data to train the SVMs and 10% to train the sigmoid, and each cross validation round uses 90% of the data again, so each SVM uses $0.9 \times 0.9 \times 12500 = 10125$ training images). To learn the SVMs for all concepts with libSVM took 11h40m. In comparison, TagProp is fast to train as it learns one set of parameters shared among all concepts, and does no require cross-validation to set regularization parameters. To learn the TagProp model using distance-based weights using $K = 200$ neighbours and including the sigmoid, takes 1m47s for the 38 concepts and 12.500 training images. These run-times exclude visual feature extraction, and computation of pairwise distances.

4.3 Learning concepts with Flickr tags

In this section we investigate how we can use the Flickr tags to learn our models. First, we use the tags as an additional feature to train SVM classifiers, and the manual annotation to define the relevance of the training images for each concept. Second, we use the tags instead of the manual annotation to define the relevance of the training images.

4.3.1 Using Flickr tags as features

To use the Flickr tags as features, we endow each image with a binary vector of length 457 indicating the absence/presence of each tag. Since we know that the Flickr tags are noisy, we also consider a 457 long vector with the tag relevance predictions of TagProp with distance-based weights. Since our implementation of TagProp exploits the sparsity of the annotations, the model is learned in 2m35s (compare to the 1m47s when learning from the 38 concepts).

We train the SVM models for each concept as before, but using the different features, and combinations thereof. For the new features, the tags and their TagProp predictions, we use a linear kernel. When combining different features we average the kernel matrices, which is equivalent to concatenating the corresponding feature vectors.

From the resulting AP scores in Table 2 we can make the following observations. On average, using the Flickr tags as features or their TagProp predictions performs similar (43.7% and 43.6% respectively), and their combination works surprisingly well (58.8%), which is better than the visual features alone (52.4%). Adding the Flickr tags as features helps significantly in all settings, whether we only use visual fea-

	animals	baby	baby*	bird	bird*	car	car*	clouds	clouds*	dog	dog*	female	female*
v	48.1	13.8	19.1	17.4	24.8	35.2	52.0	85.3	77.8	30.4	34.8	56.7	54.2
p	39.3	6.0	6.8	8.5	11.3	23.4	32.8	77.4	70.6	16.9	19.2	52.1	45.7
t	55.1	27.1	31.1	44.1	54.8	25.2	34.2	49.6	37.6	62.8	65.6	46.6	42.6
v+p	48.9	15.1	20.7	19.2	26.3	36.5	53.6	85.3	77.9	30.9	34.8	57.2	54.9
v+t	65.4	34.9	44.4	52.2	64.4	44.9	64.2	85.7	78.5	70.1	75.0	62.0	60.7
p+t	62.2	32.7	40.3	49.6	61.0	38.1	55.7	78.5	72.7	66.9	70.7	59.9	57.4
v+p+t	**66.1**	**35.9**	**45.4**	**53.5**	**65.2**	**46.4**	**65.7**	**85.8**	**78.8**	**70.4**	**75.3**	**62.6**	**61.3**
	flower	flower*	food	indoor	lake	male	male*	night	night*	people	people*	plant	portrait
v	53.3	64.9	49.7	75.4	26.8	50.2	42.3	65.7	55.7	79.8	76.1	79.9	69.1
p	41.0	50.8	41.9	69.7	19.1	47.8	36.8	60.1	46.1	75.5	68.3	74.7	57.2
t	52.1	57.9	39.9	59.7	19.2	41.6	30.5	39.9	29.7	72.2	62.3	61.7	46.7
v+p	53.7	65.2	51.2	76.2	27.3	50.7	42.9	66.4	56.5	80.1	76.3	80.1	69.3
v+t	66.3	76.8	62.2	78.8	33.2	55.7	49.0	70.1	**60.9**	85.2	81.5	82.5	73.0
p+t	62.3	71.9	57.7	75.9	30.1	54.1	42.6	67.2	53.2	83.1	77.3	79.1	66.2
v+p+t	**67.0**	**77.2**	**63.4**	**79.7**	**34.3**	**56.8**	**49.9**	**71.0**	**60.9**	**85.4**	**81.7**	**82.9**	**73.3**
	portrait*	river	river*	sea	sea*	sky	struct.	sunset	transp.	tree	tree*	water	Mean
v	69.1	24.7	6.9	57.9	29.2	89.2	78.3	67.9	45.2	68.1	54.8	62.7	52.4
p	56.7	15.1	2.4	49.0	18.8	84.4	76.4	60.3	38.4	60.1	41.1	54.6	43.6
t	46.2	29.5	4.5	43.6	14.0	67.5	69.8	39.7	33.0	39.5	29.1	53.6	43.7
v+p	69.2	25.7	7.0	57.9	29.2	89.4	78.8	68.2	45.6	68.2	55.1	63.1	53.0
v+t	72.8	39.5	**16.4**	66.4	35.5	90.4	81.8	68.4	54.2	69.7	59.3	73.7	63.3
p+t	65.8	35.8	11.1	62.3	29.2	86.8	81.4	63.5	50.4	64.1	49.5	70.1	58.8
v+p+t	**73.1**	**40.2**	16.3	**66.4**	**36.8**	**90.7**	**82.6**	**69.1**	**55.0**	**70.4**	**59.9**	**74.2**	**64.0**

Table 2: Comparison in terms of AP of SVM models that use visual features (v), Flickr tags (t), and their predictions using TagProp (p), and combinations of these.

tures, TagProp features, or both. Adding the TagProp features when the visual features are already used helps little, whether or not the Flickr tags are also used. We conclude that the TagProp features form a compact and interpretable image representation, capturing a significant amount of the information in the visual similarities used to compute them.

4.3.2 *Using Flickr tags instead of manual labels*

Next, we consider learning our models directly from the Flickr tags instead of the manual concept annotation. We evaluate the learned models using the manual annotations of test images for the 18 concepts (annotated in the non-strict sense) that also appear among the 457 Flickr tags.

For TagProp we directly used the relevance estimates from the model we trained on all 457 Flickr tags, i.e. to rank the images for the concept 'animal' we use the relevance estimates for the Flickr tag 'animal'. For SVM models we replaced the manual concept annotations with an annotation based on the absence or presence of the corresponding tag and proceed as before.

In the first three columns of Table 3 we show the performance obtained using TagProp and SVMs using the visual features only. As expected, for all concepts the performance drops significantly when learning from the noisy Flickr tags instead of the manual concept annotations. However, for all concepts performance is still significantly above chance level. Perhaps surprisingly, in this case the TagProp models perform better than the SVM classifiers in terms of average AP, BEP, iAP, and iBEP. A possible explanation for these results lies in the noise in the training labels: as TagProp has less parameters it is less likely to over-fit to this noise. This is also coherent with the fact that the distance-based weights perform better than rank-based weights in this case.

Finally, we also consider using the tag-based features when learning SVMs from the Flickr tags. Note that in this case we should exclude the tag from which we are training from the feature set, otherwise we would obtain degenerate classifiers that uses a Flickr tag as perfect predictor for itself. Apart from this issue, we train our SVM classifiers as before and present the AP scores in the remaining columns of Table 3. Generally we see the same effect of feature combinations, except that in this case the combination of the Flickr tags and the TagProp features performs worse than using the visual features alone. When using all feature sets the performance is comparable to that of the distance-based TagProp model in terms of mean AP, but still worse in terms of BEP, iAP and iBEP.

We note that the SVM approach might be improved using other approaches to combine the visual and tag-based kernels [7]. Similarly TagProp might be further improved by exploiting the Flickr tags to define the neighbour weights.

5. CONCLUSION

We have presented TagProp, a weighted nearest neighbour model for image annotation, and evaluated performance on the MIR Flickr set. We compared to SVM classifiers learned per concept, and considered both the use of manual annotations and Flickr tags to learn our models. In our experiments we show that TagProp can successfully combine different similarity measures between images. This is consistent with our earlier findings on other data sets [5, 9]. Using rank-based and distance-based weights yields comparable performance, and for either definition of the weights the addition of a per-word logistic discriminant model significantly improves performance.

In our comparison between TagProp and SVM classifiers we found SVMs to perform better when trained from precise

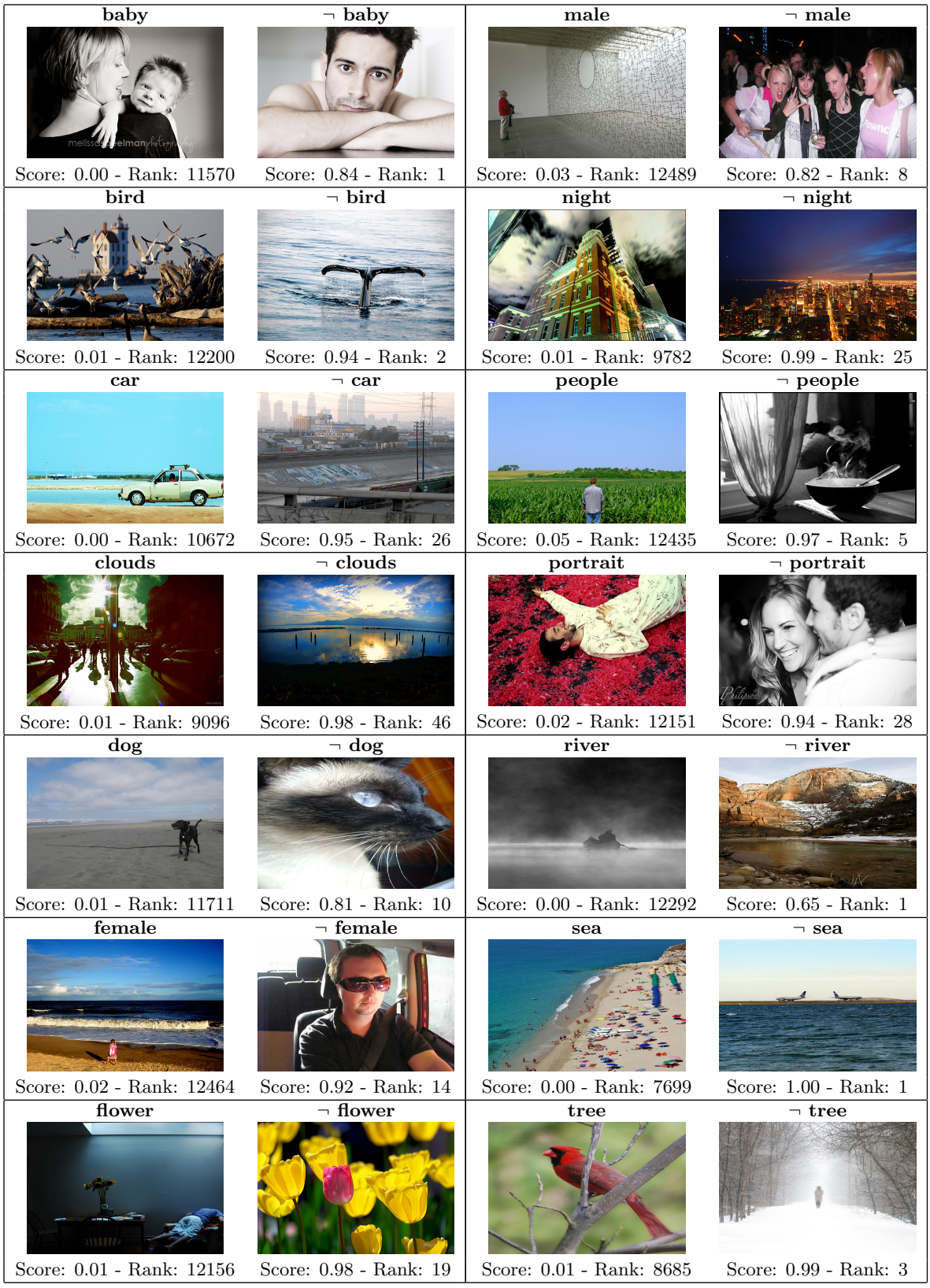

Figure 5: Lowest scoring positive and highest scoring negative example for concepts, for each image the relevance estimate and ranking among the 12500 test images is given. The 14 concepts with 'strict' relevance annotations were used. Relevance estimates were generated using TagProp with distance-based weights.

AP	Dist	Rank	SVM v	SVM p	SVM t	SVM v+p	SVM v+t	SVM p+t	SVM v+p+t	Random
animals	29.3	24.2	25.2	20.1	19.9	9.1	30.0	24.3	**32.2**	12.9
baby	7.9	7.3	**8.3**	2.6	2.0	7.1	5.6	4.3	5.9	1.1
bird	12.3	11.5	10.1	5.0	11.7	12.3	21.6	13.3	**23.0**	3.0
car	**27.5**	26.7	26.2	16.6	7.3	26.4	25.4	17.0	26.2	5.0
clouds	**66.9**	66.3	61.9	45.7	33.2	63.2	56.9	51.4	58.6	14.5
dog	17.0	18.9	20.1	9.6	23.2	22.3	34.8	27.4	**35.2**	2.7
flower	37.3	37.0	42.0	25.5	31.9	43.1	47.1	37.0	**48.1**	7.4
food	40.4	40.3	39.2	23.7	9.4	40.8	47.1	32.4	**47.9**	4.0
lake	19.1	**21.3**	19.1	8.9	9.7	19.9	17.5	14.9	19.3	3.0
night	**55.7**	54.3	45.4	35.0	24.2	44.1	45.9	42.0	46.1	10.3
people	**59.0**	57.9	52.2	47.5	47.5	53.5	53.4	51.4	55.1	41.3
portrait	39.3	39.4	43.0	23.0	27.9	42.5	**43.6**	37.5	42.3	15.6
river	17.0	16.0	15.7	9.4	12.5	15.9	20.0	15.5	**21.7**	3.7
sea	**45.0**	42.2	33.1	21.1	23.4	31.3	39.0	28.6	38.5	5.3
sky	**69.7**	68.8	64.3	56.5	49.2	64.4	61.9	58.7	62.8	31.0
sunset	54.3	55.1	59.7	47.7	23.9	**61.1**	56.0	47.6	56.7	8.4
tree	**44.2**	40.4	33.5	27.2	25.0	34.1	33.5	32.0	34.6	18.3
water	**49.4**	46.2	38.4	22.6	31.9	39.6	42.1	35.8	42.6	13.1
Mean AP	38.4	37.4	35.4	24.9	23.0	35.0	37.9	31.7	**38.7**	11.1
Mean BEP	**39.7**	38.2	36.4	27.3	24.6	36.0	38.4	33.4	39.2	11.1
Mean iAP	**47.3**	46.3	44.2	36.4	32.0	44.5	45.0	42.5	46.2	5.6
Mean iBEP	**36.5**	35.4	33.3	24.4	19.3	33.0	33.6	30.7	34.0	5.6

Table 3: Performance when training from user-tags, using TagProp (distance-based and rank-based weights), and SVMs with different feature sets: visual (v), TagProp tag predictions (p), and Flickr tags (t).

manual annotations, but to perform worse when using the noisy Flickr tags as training labels. We think this is due to the fact that TagProp has far fewer parameters than the SVM approach, therefore TagProp is less suited to learn a complex decision boundary from precise manual annotation, but also less likely to over-fit to the noisy labels given by the Flickr tags. Using both forms of supervision, including Flickr tags as features improves the performance of SVM classifiers, in particular when learning from the manual annotations. Interestingly, when learning SVMs from the manual annotations, the combination of the Flickr tags and their TagProp predictions yield a performance above that given by the visual features alone.

In future work we want to address the modeling of the correlation between the presence of annotation terms, which is currently not taken into account in our annotation models. Furthermore, we want to explore learning models using objective functions that are geared towards optimising image annotation performance rather than retrieval, see e.g. [2] for recent ideas along these lines. The main difference with optimization for retrieval is that it is not necessary that relevant terms obtain a high score, but that it is sufficient that their score is higher than the score for non-relevant terms.

6. REFERENCES

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.

[2] S. Bucak, P. Mallapragada, R. Jin, and A. Jain. Efficient multi-label ranking for multi-class learning: application to object recognition. In *ICCV*, 2009.

[3] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3):394–410, 2007.

[4] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proceedings Internet imaging (SPIE)*, volume 5304, 2004.

[5] M. Douze, M. Guillaumin, T. Mensink, C. Schmid, and J. Verbeek. INRIA-LEARs participation to ImageCLEF 2009. In *Working Notes for the CLEF 2009 Workshop*, 2009.

[6] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[7] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

[8] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.

[9] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[10] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, 2004.

[11] M. Huiskes and M. Lew. The MIR Flickr retrieval evaluation. In *ACM MIR*, 2008.

[12] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *PAMI*, 32(1):2–11, 2010.

[13] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[15] J. Li and J. Wang. Real-time computerized annotation of pictures. *PAMI*, 30(6):985–1002, 2008.

[16] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*,

42(2):218–228, 2009.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008.

[19] T. Mei, Y. Wang, X. Hua, S. Gong, and S. Li. Coherent image annotation by learning semantic distance. In *CVPR*, 2008.

[20] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: Constraining the latent space. In *ACM Multimedia*, 2004.

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[22] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *ACM SIGKDD*, 2004.

[23] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *ECCV*, 2006.

[24] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. In *Workshop on Multimedia Data Mining ACM SIGKDD*, 2008.

[25] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, pages 2126–2136, 2006.

[26] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73(2):213–238, 2007.